



CODING PROPOSAL 2013

This proposal is an overview for the possible integration of the skill sets associated with coding and computational thinking into St. Anne's-Belfield's K-12 curriculum. It is presented as a draft with the intent to stimulate dialogue around how this can occur in a seamless and systematic manner. Included in this proposal are: the purpose, rationale, implementation considerations, desired outcomes, timeline including phased implementation and documentation of resources.

CODING PROPOSAL 2013

OCTOBER 1, 2013

Purpose

In the current educational setting, the impact of technology on both the teacher and learner has far outpaced the ability of any given organization to maintain itself on the “cutting edge”. From initiatives emphasizing a device in every student’s hands, to training teachers in the use of the latest gizmo to enhance the learning environment, the focus on hardware has been the greatest determinant of an organization’s ability to utilize technology. The devices currently in our classrooms will not be the devices that our learners will be using in the future. A change in thinking about how technology is used in our curricula has to occur and educators must develop a different perspective with regards to how to integrate the computer science objectives into content areas.

The purpose of this document is to provide an overview that tackles the issue of integrating technology into our school, not from a hardware or software standpoint, but from a broader conceptual (thinking) point of view. This perspective requires first agreeing upon a rationale as to why we need to *knowingly* teach computational thinking and coding within the existing curriculum. The next step would be to develop a scope and sequence that provides the content and experiences our learners require in order to apply these skills to real life questions in a meaningful manner. The final step of this process would be to have agreed upon outcomes across the schools that are imbedded and purposeful within each content area. This would then foster, starting from the early grades, the desired thinking skills our learners will require to truly harness the power that the rapid enhancement in computer sciences is affording.

“I think everybody in this country should learn how to program a computer because it teaches you how to think.” - Steve Jobs

“Learning to write programs stretches your mind, and helps you think better, creates a way of thinking about things that I think is helpful in all domains.” - Bill Gates

“Learning to code unlocks creativity and builds confidence in students regardless of age, gender, or race. Computer programming can also be a teaching tool for other subjects, from physics to French. With today’s technology, learning to code is more fun and more accessible than ever, and America’s teachers welcome it into our schools and classrooms.”

- Dennis Van Roekel
President, National
Education Association

Rationale

The development of higher level thinking skills enhances and broadens a student's ability to innovatively problem solve. The acquisition of these skills through the process of coding is limitless and provides the necessary foundation to access the power afforded through technology to impact the world around them.

Why Coding?

Coding is the process that allows for the implementation of computational thinking. It is the process of creating command statements into some sort of workable or identifiable code.

It develops:

- logical, sequential thinking in action through trial and error;
- creativity by allowing for multiple ways to address an issue;
- insights into how one thinks.

Why Computational Thinking?

Computational thinking provides the foundation from which all technology currently evolves. It emphasizes logical thinking and the use of abstraction to solve problems.

It promotes:

- a way of thinking that is based on problem solving;
- a broad solution-based approach that uses computing in an intelligent manner;
- a structured approach that revolves around intentional thinking strategies, breaking large problems into smaller pieces.

Why Computer Science?

Computer Science is the scientific approach to understanding not only how computers work, but how computation works. Computer Science teaches students design thinking, logical reasoning, and problem solving skills. It enables the student with the ability to create new technologies and adapt known technologies.

Implementation Considerations

In reviewing the literature and talking with computer science leaders, it becomes apparent that a fully integrated model of coding is not easily found in a K-12 curriculum. Computer Science courses traditionally focus on dedicated classes and are usually taught at the highest levels of education, not within the reach of our youngest users. There are many examples of specific courses taught, but rarely is it seen as a continuum from grade to grade. There are a few schools with 9-12 coding programs, a few schools with K-6 or 3-8 programs, but we have not come across a fully developed K-12 integrated model.

Often, coding is presented almost as an afterthought for students who have mastered the use of current devices and software and that exhibit a desire to go further in the field of technology. In many instances coding is taught in isolation. This manner of teaching doesn't allow for the learner to be able to apply the thinking skills across domains. In many cases teachers are not even part of the process, and the activity is provided outside of the learner's "school day" as extra curricular activities without a true connection back into the core content areas.

Desired Outcome

The intent of this proposal is to outline an action plan for creating a framework for a K-12 scope and sequence for conceptual thinking and coding. This framework would then be used by teachers to imbed coding into existing content area. The final outcome of the process would be the implementation of coding and conceptual thinking skills, which would be imbedded in the school's curriculum from kindergarten through 12th grade.

Timeline (Phases)

Phase 1 (Alpha): “Planning, Fact Finding and Review” Ongoing - 2013-2014 School Year

Phase One is the development and refining of the proposal that includes furthering our understanding of coding and conceptual thinking and continuing to develop and modify our plan based on the acquired information. This is the dynamic phase that allows us to have flexibility based upon what we learn and experience and provides an avenue for ensuring that we have the resources and foundation from which to evolve our school programs.

1. Literature review and development of a knowledge depository.
2. Research, talk with and visit schools that are currently somewhere in the process.
3. Participate in professional development opportunities (conferences, conference calls, etc.).
4. Development of an action plan that provides steps with drafts of timelines, administrative structure, associated costs, etc.

Phase 2 (Beta): “Sow the Seeds with Staff” – 2013-2014 School Year

In planning ahead it is essential that we have “buy in” by faculty and that they have had experiences in which they have seen the process and desired outcomes. In order for this to happen we would plan a series of “pilot” sessions that are:

1. Ongoing: TAC classes (team teaching) integrating coding and computational skills
 - a. Provide classes incorporating coding apps and software as opportunity arises with all teachers;
 - b. Select a few teachers to pilot a few programs with their classes as it naturally could be integrated (4/5th science – Scratch).
2. Ongoing: Teacher initiated integration of coding and computational skills
3. Intensive (boot camp): One hour a day for one week, selected teachers & students will participate in coding lessons.
 - a. The teacher and the TRC will teach in tandem.
 - b. Input from students, teachers and TRC will help to determine what was gained from participation (survey, interviews, video collage)
 - c. Teachers spend time at the end of the week to share, debrief, and expand upon the experience that they created with their students.

Phase 3 (Debugging): “Think Tank”: Continuum of Skills/Outcomes – Summer 2014

Bring leading experts in the fields of coding, conceptual thinking, mind research, etc. into our dialogue to help us develop a scope and sequence that defines a continuum of outcomes/skills. This think tank may occur at the BLC 2014 conference, or here at St. Anne’s-Belfield.

Phase 4 (Implementation): “Watering the Seeds and Watching it Grow” Design Charrette – Fall 2014

In this phase we brainstorm ways to integrate the scope and sequence into each content area and develop specific lessons. Professional development experiences will occur at this stage, and faculty will create specific implementation plans that will include any resources they think they would require. During the implementation phase, we will be collecting feedback from all participants to document and evaluate the program.

Resources

For articles on this topic, please use this link:

<https://www.diiigo.com/user/stabls/coding?type=all&sort=updated>

For list of software, apps, programming games and coding programs:

See shared google document – Coding Resources